A METHOD OF ENABLING A WIRELESS INFORMATION DEVICE TO AUTOMATICALLY MODIFY ITS BEHAVIOUR

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to a method of enabling a wireless information device to automatically modify its behaviour. The term 'wireless information device' used in this patent specification should be expansively construed to cover any kind of device with one or two way wireless communication capabilities and includes without limitation radio telephones, smart phones, communicators, personal computers, computers and wireless enabled application specific devices such as cameras, video recorders, asset tracking systems etc. It includes devices able to communicate in any manner over any kind of network, such as GSM or UMTS, CDMA and WCDMA mobile radio, Bluetooth, IrDA etc.

2. Description of the Prior Art

Enabling devices to automatically alter their behaviour depending on the environment or 'context' of the device is a compelling foundation of 'context aware' computing. 'Context' can cover device specific variables such as location, as well as end-user variables such as presence (i.e. the availability of an end-user to use the device, his mood etc.). One common strand in context aware computing is the need for the device to itself become automatically aware of its context: for example, a location aware device I typically equipped with location finding equipment, such as GPS, or the ability to acquire location information from a nearby source.

Equipping a device with location awareness enables new capabilities: for example, the device could automatically turn itself off when in a location in which device operation is hazardous (in a hospital or aircraft, for example).

One of the assumptions behind context aware computing is that the context aware device automatically acquires its context understanding. The present invention does not however deal with devices that can automatically acquire context information; as such it departs from the general thrust of development in this area. Instead, it deals with the situation in which the end-user manually inputs context type information into the device.

Further, prior art context aware systems focus on location and presence information. The present invention does not, but instead deal specifically with time sensitive information.

SUMMARY OF THE PRESENT INVENTION

In a first aspect, there is a method of enabling a wireless information device to automatically modify its behaviour, comprising the steps of:

(a) an end-user entering time sensitive information into a first application running on the device;

(b) a second application running on the device receiving data directly or indirectly from the first application, the data relating to the time sensitive information, and the second application then automatically changing the behaviour of the device appropriately in dependence on the data.

Hence, the present invention deals with the very specific situation of an end-user entering time sensitive information into an application; this, for example, may be an entry (e.g. 'meeting', 'lunch with Bob', 'travelling', 'flying' etc) against specific times in an agenda or calendar application. Then, a different application on the device can utilise that information to modify the device behaviour appropriately. For example, say the 'meeting' in the calendar application is listed to last between 10am and 11am; then, during that hour, the telephone application in the device (that enables the telephone functions of the device to be controlled) could automatically be set to a suitable profile, such as a 'silent' profile so that the device does not ring on an incoming call, but instead only vibrates. The term 'application' should be expansively construed to cover any structure of software that performs one or more functions; it hence covers elements/portions of an operating system, utilities, client components, server components etc. Particularly, the second application could be OS system services, as opposed to an application which presents a unique interface to an end-user.

The end-user could, in relation to the agenda application entry (e.g. 'meeting between 10am and 11am) select from a menu list (i.e. any kind of user interface that enables the end-user to select different options) a label to apply to the entry, the label defining the type of behaviour change to be carried out by the second application. Hence, in relation to the 'meeting', 'lunch', 'travelling' or 'flying' etc. entries, the end-user could activate a

pop-up menu of possible behaviour changes linked to that event. These might include the following selectable options:

(a) altering a telephone profile (e.g. silent, vibrate, loud, divert to voice mail etc.)
(b) altering the device ring tone
(c) altering the device user interface
(d) switching off telephone functionality
(e) switching off the device entirely
(f) switching the device to a power save mode
(g) switching off one or more items of communications hardware (e.g. GPS, IR, Bluetooth, wireless LAN modules).

Generally , the second application automatically changes the behaviour of the device appropriately in dependence on the data from the first application for a time period determined by that data, e.g. silent mode for 1 hour to coincide with the meeting duration.

Another example is that there could be a dialog near the weekdays dialog in the agenda application with pairs of times the end-user enters to put the phone into power save mode, weekday sleep times, Saturday sleep times, Sunday sleep times etc.

In one implementation, the first application sends the data indirectly to the second application via an intermediary server. The server operates as an insulation layer, separating the first and second applications (the term 'insulation layer' is a term of art in design patterns). It allows there to be one or several first applications (i.e. applications into which the user can input time sensitive information – such as (a) an agenda application into which the end-user can enter appointments, meetings, events etc. occurring at different defined times and (b) an alarm application into which the end-user can input a time he would like the device to sound an alarm). Further, the server enables any of these first applications to send data to any number of other applications that are to respond to the time sensitive information (or more specifically the data sent from the first application(s) that relate to this time sensitive information). Hence, a device might

be in sleep mode, but the power management application be set to awake to normal mode when the alarm sounds: in this case, the first application is the alarm application and the second application is the power management application. The server can also ensure that the second application is running and to launch it if necessary.

The approach of the present invention therefore provides a structured and systematic way for a device to intelligently use time sensitive information that the end-user has manually input.

If a conflict arises between the behaviour change due to the data from the first application and a different behaviour change input directly by the end-user to the first or the second application, then the different behaviour change may prevail – different types of changes may be allocated different priorities. Hence, the end-user could place the device directly back into sleep mode as or shortly after the alarm is sounding, to continue with the example above.

In one implementation, dynamically ensuring event-precedence is deterministic - meaning all events must guarantee to resolve to correct behaviour. So if one event has higher priority than another, this must be true under all circumstances ('if and only if'), and if two events have the same priority they must resolve to the same result each time.

If a conflict arises between the behaviour change due to the data from the first application and a different behaviour change input directly to the first or the second application, then a conflict resolution component determines which behaviour change prevails. A component designed to resolve conflicts can have a pre-defined list of conflict situations and the appropriate conflict resolution decision it should make given those inputs. The pre-defined conflict list may contains derivation rules, i.e. allow algebraic substitution rather that a static scenario and are thus are dynamic and will not cause compatibility problems; this facilitates third parties subscribing and publishing with the system.

An override component could also be provided to determine if a behaviour change due to the data from the first application is inappropriate and to then override that behaviour change. Hence, a device may be automatically about to wake up from sleep mode and sound the alarm as described above; but the device may also have a location sensor that informs it that it is in an aircraft and hence must not activate. In this circumstance, the wake up behaviour change would be over-ridden.

The intermediary server could provide the conflict resolution component and override component.

In addition, it is possible for the end-user to enter time sensitive information into the first application running on the device; and for a second application running on a second, different device to receive data directly or indirectly from the first application, the data relating to the time sensitive information, and the second application then automatically changing the behaviour of the second device appropriately in dependence on the data.

In this way, for example, an alarm notification set into one device could be used to trigger an action within or by an application running on a completely different device; the two devices might be connected over a wireless link. Hence, it could trigger a web camera to activate and send images to a web site.

In a second aspect, there is a wireless information device programmed to automatically modify its behaviour, the device enabling:

(a) an end-user to enter time sensitive information into a first application running on the device;

(b) a second application running on the device to receive data directly or indirectly from the first application, the data relating to the time sensitive information, and the second application then automatically changing the behaviour of the device appropriately in dependence on the data.

DETAILED DESCRIPTION

The present invention can be implemented using a time sensitive information "insulation layer" in the wireless information device SymbianOS from Symbian Limited; the insulation layer separates and hence decouples applications and any system servers from a source application that an end-user initially enters time sensitive information into (e.g. an agenda or calendar application; an alarm application etc). Hence, an agenda application could provide time sensitive information to the insulation layer (a "change server" in a Symbian OS implementation). This server insulation layer then gives access to the time sensitive information data to any client (e.g. an application or system server) that needs it over a generic API. The client need not be limited to a client on the wireless information device itself, but can reside on a completely different device that does not have its own system for an end-user to input time sensitive information into. Hence, the device into which an end-user enters the time sensitive information can be used to control the behaviour of a completely different device over pre-set times.

There are many advantages to a time sensitive information insulation layer approach:

- Application developers and network operators can develop time based applications without needing to understand the specifics of any time aware application; they merely need to be able to use the API.
- Any application or resource on the device that needs time sensitive information entered by an end-user can now readily obtain it; this time sensitive information is made available across the entire OS. Hence, alarm notifications are no longer limited to just the stand alone alarm application itself, but can be used to initiate actions in any other application.
- The time sensitive information can be readily supplied to different system servers – e.g. a system server which decided which power mode to adopt could do so depending on the time sensitive information – off mode at night; sleep mode in the evening etc.
- A publish and subscribe API can be used to enable any such authorised server, to subscribe to time sensitive information published by any of the time aware applications (e.g. a calendar application).

A more detailed example follows: Assume that the end-user inputs into the agenda application that he has a meeting between defined times; the time of the meeting is then held in the agenda application's data. Assume then also that the device profile setting is in the phone application. This determines the ringing/call diversion behaviour of the device.

The invention provides an independent mechanism that can allow data communication between the 'source' agenda application and the 'destination' phone application. In the present invention, the manner in which the destination application behaves can be dictated in advance by the source application(s). The behaviour in the above example is to change the device profile to Meeting mode while during the meeting and back to whatever it was after the meeting ends.

The agenda application needs to communicate with phone application: in essence, the agenda application needs to say to the phone application: 'switch to meeting profile' at a specific time and switch back afterwards. (The phone application is programmed with the profiles and different phone companies will do different profiles. Hence, the agenda application would have to ask the phone application for what profiles it uses so the agenda application can ask the user which profile to use against each agenda entry type).

All applications have UIDs in SymbianOS so that is possible to have a change server with which all applications (including system services) register, to allow applications to send data/messages to other applications without actually making the destination application become the active application.

There are several practical issues to consider:

1. If the meeting finishes early, the user may manually change the profile; this should probably override the agenda application trying to changing it back when the meeting officially ends according to its data.

2. The user could also use a menu option to say the meeting has ended now and the end of the meeting time is automatically updated in the agenda to the current time.

3. The clock alerts the agenda application about the meeting. The agenda application then sends a message to the change server that consists of a UID (for the phone application) and function number in a message for the phone application to say which profile it is currently using. Then the agenda application sends a message to the change- server that consists of a UID (for the phone application), message UID and the message in a package descriptor that consists of a data structure of one profile switching message. It is best not to send the phone application a switch-back-at-11 message as this relies on the phone application both having to remember the current profile and implementing a request to some alarm functionality. So at 11am, using the 10am to 11am meeting example, the agenda application then sends a message to switch back to the profile that the agenda application has stored; this message may be of a switch-back priority. If the user (or something of higher priority) modifies the profile during the meeting, then this switch-back message is ignored.

Also, setting back a profile to something previously set may not make any sense when we get to that time (e.g. there may have been an intervening higher priority event).

Conflict resolution is a particular feature of this approach: it could be that the source application (that has application specific information about the situation) or the intermediary change server (that has general device specific information about the state of the device). The change server would generally be best placed to perform priority conflict resolution or mediation. For example, if the end-user starts having a meeting in an airport lounge and continues as he gets on the plane: The phone is going to be told to switch off its air interface radio by the airport which is more important than changing profiles. If the device still works as a PDA, it can still make of the change profile settings so that the phone is in the appropriate mode when the end-user gets off the plane.